

논문 2014-09-02

항공 시뮬레이터를 위한 OpenSceneGraph기반의 고도 정보 구현 방안

(Implementation of Altitude Information for Flight Simulator in
OpenSceneGraph)

이 충 재, 김 중 범, 김 기 일*

(ChungJae Lee, JongBum Kim, Ki-Il Kim)

Abstract : When it comes to develop flight simulator, HAT (Height Above Terrain) is required to provide altitude information to the pilot who learns how to control an airplane in landing and takeoff situation. However, there might be inconsistent problem between real terrain and simulation information since current implementation of HAT simply depends on center of gravity point on the airplane. To overcome mentioned problem, in this paper, we propose how to obtain more accurate altitude information than existing scheme by making use of HAT and HOT (Height Of Terrain) information of landing equipments according to movement of the airplane. Moreover, we demonstrate the accuracy of the proposed scheme through new flight simulator developed through OSG(OpenSceneGraph) by taking example of terrain information for domestic airport.

Keywords : Flight simulator, HAT/HOT, OpenSceneGraph

I. 서론

공기 조종을 위한 시뮬레이터 소프트웨어를 개발하기 위한 비주얼 시스템은 크게 디스플레이 장치와 영상 발생 장치(Image Generator, IG)의 두 가지 하부 시스템에서 동작한다[1]. 이중 영상 발생 장치의 비주얼 시뮬레이션 기술은 탑승자의 시야각과 현재 조종기의 입력값에 따른 항공기 움직임을 실제 항공기와 동일하게 묘사하기 위하여 사용된다. 이를 위해서는 시간 변화에 따른 시뮬레이션 환경

변화가 필요하며 비행에 관련된 정보를 처리하기 위한 소프트웨어 기술이 요구된다.

이러한 소프트웨어 기술 중 HAT와 HOT는 시뮬레이션 환경에서 탑승자의 이착륙 시 공항의 정확한 고도 정보를 제공함으로써 사실적인 이착륙 훈련에 필수적인 요소이다. 시뮬레이션 환경에서 HAT와 HOT를 통하여 현재 항공기의 위치 정보와 더불어 정확한 고도 정보를 탑승자에게 인시킴으로써 사실적인 비행 훈련에 몰입이 가능하다.

하지만 현재 실제 개발되어 있는 상용 비행 시뮬레이터의 경우 지형의 삼각망 정보를 기준으로 Bounding Box를 이용한 충돌 감지 기능을 구현하므로 항공기의 무게 중심점을 기준으로 HAT 정보를 구현하거나 일반적인 현재 항공기의 고도를 HAT로 사용한다. 이는 비행 시뮬레이션의 가장 중요한 이착륙 훈련 시 Bounding Box의 고도가 실제 고도와 동일하지 않을 경우 시뮬레이터가 정확하게 동작하지 않는 문제점이 발생한다. 이러한 상황에 대비하여 현재 시뮬레이터에서는 항공기와 지형의 충돌상황에 대한 처리를 위하여 오브젝트간의 충돌을 감지하는 추가적인 모듈의 구현하고 있는데 이는 개발의 복잡성을 증가시키게 된다.

*Corresponding Author (kikim@gnu.ac.kr)

Received: 30 July 2013, Revised: 7 Oct. 2013,

Accepted: 28 Nov. 2013.

C.J Lee, J.B. Kim, K.I. Kim: Gyeongsang National University, Engineering Research Institute

※ 본 논문은 교육부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과 (No. 2011-02-기-01-010) 및 미래창조과학부 및 정보통신산업진흥원의 고용계약형 SW석사과정 (NIPA-2013-HB301-13-1004) 연구결과임

하지만, 이러한 문제점에도 불구하고 아직까지 정확한 HAT를 확보하기 위한 추가적인 연구는 이루어지고 있지 않고 있으며 HAT의 구현에 대한 내용은 각 시뮬레이터에서 개별적으로 구현되어 그 정확성을 검증하는데 한계가 있다. 이러한 문제를 해결하기 위하여 본 논문에서는 무게 중심점 이외에 항공기의 이착륙 시 실제 활주로와 접촉하는 부분인 세 개의 착륙 장치에 대한 HAT와 HOT 정보를 구현하는 방안을 제시한다. 제안된 방법의 경우 항공기의 삼축 움직임에 따라 변화하는 HAT 정보를 연산하게 되는데 이는 비행역학과 연관이 되어 있으므로 실제 계산은 기종에 따라 달라진다. 따라서, 본 논문에서는 이를 계산하기 위한 정형화된 절차를 정의하고 실제 개발하는 방법을 소개하고자 한다. 제안된 방법을 적용할 경우 기존 방법과 달리 추가적인 오버젝트간의 충돌감지를 위한 모듈에 필요하지 않을 뿐만 아니라 보다 정확한 고도 정보의 구현이 가능하다.

본 논문의 구성은 다음과 같다. 1장에서는 본 논문의 배경을 설명하였고 2장에서는 본 논문의 개발 대상인 HAT, HOT의 개발에 대한 연구 및 문제점을 분석하였다. 3장에서는 구현을 위한 소프트웨어 구현 방법 및 개발환경을 설명하였고 4장에서는 이에 대한 검증을 수행하였다. 마지막 5장에서는 결론을 제시한다.

II. 관련연구

항공 조정 시뮬레이터에서 사용되는 DTED (Digital Terrain Elevation Data)는 특정 범위에 대한 지형의 고도 정보를 위성 혹은 항공기를 이용하여 관측한 고도 정보점을 면으로 연결한 데이터이며 이를 이용하여 지형정보를 제공하게 된다. 하지만, 실제계의 모든 지형 사물을 묘사하는 것은 현실적으로 불가능하기 때문에 시뮬레이션 동작중 항공기의 현재 고도와 지형의 고도 정보는 실제계와 완벽하게 동일 할 수 없다. 따라서 항공기 시뮬레이터 개발 시 탑승자의 현재 시야에서 고도 인지와 이착륙 상황에 대한 정확한 고도 정보 파악을 위한 HAT와 HOT 구현은 필수 사항이다. 그림 1은 HAT와 HOT에 대한 정의를 보여준다.

그림 1에서 보이는 바와 같이 현재 항공기의 위치에서 지형과 일직선상에 있는 지형과의 거리를 HAT라 정의하며, 동일 지점 상 지형의 고도를 HOT라 한다. HAT와 HOT는 시뮬레이터 탑승자의

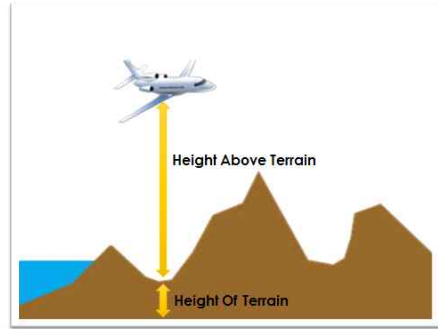


그림 1. HAT와 HOT 정보
Fig. 1 HAT and HOT Information

시뮬레이션 상황에서 항공기가 진행하는 방향의 좌우 범위와 전방 범위만큼의 영역을 주기적으로 분석함으로써 충돌 상황을 경고하는 기능을 제공하는 동시에 항로 단면도 분석을 위한 데이터로도 사용된다[2].

이러한 HAT와 HOT는 연구보다는 개발에 초점이 맞추어져 있다. 이는 HAT를 위한 guard position의 개수를 줄이기 위한 이론적 방법을 소개한 참고문헌 [3] 이후 추가연구가 제시되지 않고 있다. 반면 HAT와 HOT를 개발하는 방법에 대한 소개는 각 상용시뮬레이터의 정보를 통하여 알 수 있다. 예를 들어 Flight Simulator[4]는 HAT 연산을 위하여 지형의 고도 정보를 추출하여 HOT를 연산하며, 이를 기준으로 둘 사이의 차이를 연산하여 HAT를 구한다. 이는 항공기의 무게 중심점에서의 고도 정보를 이용하여 구현하므로 실제 지형과 맞는 부분인 착륙 장치에 대한 HAT 정보는 포함하지 않는다. 또한 Flight Gear[5]와 X-Plane[6]의 경우도 Flight Simulator와 동일하게 HAT를 연산한다.

하지만, 이러한 무게 중심점 기반의 구현방법은 다음과 같은 문제점이 존재한다. 1) 첫째, 항공기가 활주로 위에 정렬한 상황에서는 HAT 값이 0이 아닌 무게 중심에서 착륙 장치 사이의 거리 값이 표시된다. 이는 착륙 시 사용자에게 일반적인 지형의 고도 정보를 제공할 경우 문제가 되지 않지만 지형과의 거리를 고도 정보로 표시할 경우 착륙 시 고도가 0이 아닌 값을 표시하는 문제가 발생한다. 2) 둘째, 충돌감지를 위한 추가적인 방법이 필요하다. 이는 무게 중심을 기준으로 HAT를 연산할 경우 Bounding Box를 이용하여 지형의 충돌 감지 모듈을 구현해야 한다. 이러한 지형 충돌 감지는 지형의

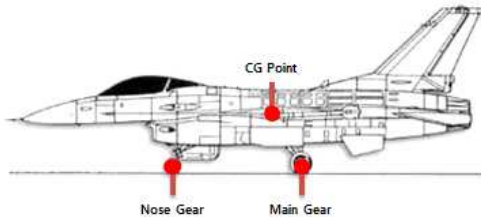


그림 2. HAT와 HOT를 위한 장치

Fig. 2 Airplane Equipments for HAT/HOT

고도 정보에 충돌 시 더 이상 고도를 하강하지 않도록 하는 지형의 삼각망에 Bounding Box를 생성한다. 이는 지형의 연산량 증가와 함께 시뮬레이터 개발의 복잡성을 증가시킨다. 따라서 무게 중심점 이외에 착륙 장치의 HAT를 연산 시 앞서 언급한 단점에 대한 해결책과 동시에 보다 정밀한 시뮬레이션 환경을 제공한다.

III. OSG 기반의 HAT/HOT 구현

본 절에서는 기존의 문제점을 해결하기 위하여 새롭게 제안된 HAT와 HOT를 구현방안을 설명한다. 이를 위해 본 논문에서는 네 지점에 대한 HAT와 HOT를 연산한다. 개발을 위해서 OSG를 사용한다. OSG는 오픈소스 라이브러리로서 3차원 그래픽 처리를 위해 OpenGL을 기반으로 OSG는 3차원 공간상에 존재하는 다수의 가상물체 및 멀티미디어 객체를 효과적으로 관리하기 위해 사용되는 계층적 트리 형식의 자료구조인 Scene Graph 이론에 기반한다[7-9]. OpenGL의 경우 저수준의 단계부터 구현이 가능하지만 이로 인해 개발 시 개발 속도가 상대적으로 느리다. 하지만 OSG의 경우 OpenGL을 기반으로 개발자의 편리성을 위해 제작된 라이브러리로서 단시간에 고수준의 3차원 그래픽 처리 소프트웨어 개발이 용이하다. 따라서 본 논문에서 구현하는 연구용 비행 시뮬레이션의 개발에 용이하다. 그림 2는 본 논문에서 구현한 연구용 비행 시뮬레이션 상에서 필요한 HAT와 HOT 정보이다.

그림 2에서 알 수 있듯이 항공기의 무게 중심점(Center of Gravity Point, CG)과 전륜 착륙 장치(Nose Gear), 두 개의 주 착륙 장치(Main Gear)에 대한 정보를 이용하여 네 지점의 HAT와 HOT 정보를 구한다. 이는 항공기가 지상에 있을 경우 실제 항공기의 지면에 닿는 부분은 무게 중심점이 아닌 네 개의 착륙장치이므로 무게 중심점의 HAT이외에

착륙장치의 HAT연산이 필요하다. 하지만 다수의 항공기별로 해당 고도값이 서로 다르기 때문에 본 논문에서는 한국항공우주산업의 T-50 항공기를 기준으로 하며 해당 HAT와 HOT를 계산한다. HAT와 HOT를 구현하기 위해서는 다음 단계를 거친다.

- 1) 첫 번째로 호스트 상에서 항공기의 현재 상태 정보를 전송한다. 항공기의 상태 정보는 항공기의 위경도 위치 정보와 고도, 그리고 삼축 자세 정보를 포함한다. 호스트로부터 전송된 데이터는 영상 발생 장치에서 수신되며, 수신된 위경도 좌표 체계는 (수식 1)을 통하여 OSG 좌표 체계로 변환 된다. OSG 좌표 정보로 변환하는 이유는 본 논문에서의 영상 발생 장치는 OSG를 이용하여 구현하기 때문에 수신된 위경도 좌표 체계 사용이 불가능하다. 따라서 OSG 상에서 사용되는 좌표 체계로 변환해야 하므로 수식 1의 수식에 따라 위경도 좌표를 OSG 좌표 체계로 변환한다.

$$\begin{aligned}
 OSG_{Coordinate_Y} &= (Latitude - Latitude_{OffSet}) \\
 &\quad * Latitude_{MeterToDegree} \\
 OSG_{Coordinate_X} &= (Longitude - Longitude_{OffSet}) \\
 &\quad * Longitude_{MeterToDegree} \\
 Latitude_{MeterToDegree} &= Constante\ in\ Latitude \\
 Longitude_{MeterToDegree} &= Constante\ in\ Latitude \\
 &\quad * \cos(Longitude_{OffSet} \\
 &\quad * \pi / 180) \\
 * OSG_{Coordinate_Y} &: OSG\ 좌표로\ 변환된\ 경도 \\
 OSG_{Coordinate_X} &: OSG\ 좌표로\ 변환된\ 위도 \\
 Latitude_{OffSet}, Longitude_{OffSet} &: 위경도\ 초기\ 값 \\
 Latitude_{MeterToDegree} &: 경도\ Radian\ to\ Meter\ 변환\ 값 \\
 Longitude_{MeterToDegree} &: 위도\ Radian\ to\ Meter\ 변환\ 값
 \end{aligned}$$

수식 1. OSG 좌표 변환

Formula 1. Convert to OSG coordinate

수식 1에서 $Longitude_{MeterToDegree}$ 와 $Latitude_{MeterToDegree}$ 는 각도값을 미터로 변환하기 위한 값을 의미한다. 또한 OSG 좌표 체계의 중심점 값을 위한 위경도 초기 좌표 위치를 입력한다. 마지막으로 $OSG_{Coordinate_Y}$ 과 $OSG_{Coordinate_X}$ 는 입력받은 현재의 위경도 값을 OSG 좌표로 변화하는 수식이다.

- 2) 앞선 1)의 과정을 통하여 변환된 위치 정보는 현재 항공기의 무게 중심에 대한 위치 정보이다. 본 논문에서는 네 지점의 위치 정보를 구현하기

표 1. 네 지점의 위치 정보

Table 1. Coordinate information of four point

	좌표(X, Y, Z)
CG	(0, 0, 0)
Nose	(0.021, 4.698, -1.962)
Main1	(-0.973, -0.488, -1.909)
Main2	(0.973, -0.488, -1.909)

때문에 무게 중심 위치 정보를 바탕으로 나머지 세 지점의 위치 정보를 구한다. 표 1에서는 예제로 구현된 네 지점의 위치 정보를 보여준다. CG는 항공기 무게 중심점을 의미하며 항공기 무게 중심점을 기준으로 나머지 세 지점과의 떨어진 거리를 미터로 표시한다. Nose는 전륜 착륙 장치의 지면과 맞닿는 부분을 의미하며 Main 1,2 는 후륜 착륙 장치를 의미한다.

3) OSG 좌표 형식으로 변화된 데이터의 HAT 연산은 다음과 같다. 우선 시뮬레이션 환경은 3차원이며 항공기 움직임 역시 삼축을 움직이기 때문에 항공기의 자세가 변경될 경우를 고려하여야 한다. 그러므로 무게 중심점을 제외한 나머지 세 지점의 삼축 정보를 연산한다. 위의 과정을 거쳐 네 지점의 위치 정보를 OSG 형식의 좌표 체계로 변경한다. 이후 HAT와 HOT를 연산하기 위한 지형 정보 노드를 구하고 검증을 위해 실제 지형 데이터를 사용한다. 사용된 지형은 한반도의 일부분 예천공항 반경 50Km 주변 지형이다. 이 지형은 HAT와 HOT 연산을 위하여 computeHeightAboveTerrain 함수에서 3개 파라미터 중 첫 번째 인자로 사용한다. 그림 3은 computeHeightAboveTerrain 함수 원형이다.

computeHeightAboveTerrain 함수는 OSG에서 제공하는 많은 라이브러리 중 osgsim 라이브러리에 포함된 함수이다. 세 개의 파라미터를 가지며 첫 번째 파라미터는 지형의 노드 정보를 입력 받는 파라미터이다. 두 번째 파라미터는 OSG 좌표를 입력 받는 파라미터로써 수식 1에서 변환한 위경도 좌표 정보를 입력 받는다. 마지막 세 번째 파라미터는 osgNodeMask로써 적용한 노드 정보를 장면에 표시하는 것에 관한 유무를 지정하는 파라미터이다. 본 논문에서는 첫 번째 파라미터로 입력받은 지형 정보를 장면에 표시해야 하기 때문에 -1 값을 입력한다. 세 개의 파라미터를 입력 받은 computeHeightAboveTerrain 함수는 최종적으

```

struct HAT
{
    HAT(const osg::Vec3d& point);
    _point(point),
    _hat(0,0) { }
    osg::Vec3d _point;
    double _hat;
};

typedef std::vector<HAT> HATList;

double _lowestHeight;
HATList _HATList;

osg::ref_ptr<DatabaseCacheReadCallback>
    _dcr
osgUtil::IntersectionVisitor
    _intersectionVisitor;

```

그림 3. computeHeightAboveTerrain 함수 원형
Fig. 3 computeHeightAboveTerrain Function

로 그림 1에서 설명한 내용처럼 입력받은 지형 노드 정보와 위경도 좌표 정보를 OSG 좌표 정보로 변환한 정보를 바탕으로 HAT와 HOT를 연산하여 리턴한다. 수식 2는 HOT 연산 과정을 보여준다.

$$\text{Altitude}_{\text{HOT}} = \text{Altitude}_{\text{Origin}} - \text{Altitude}_{\text{HAT}}$$

수식 2. HOT 연산

Formula 2. Operation of HOT

수식 2의 HOT 연산은 앞에서 연산된 HAT를 이용하여 구하며 입력받은 항공기의 현재 고도와 HAT의 차를 이용하여 구한다.

IV. 구현 검증

본 절에서는 실제 구현되어 연산된 HAT와 HOT는 정확한 값인지 판단하기 위한 검증 방법을 소개한다. 호스트 컴퓨터는 항공기의 항공 역학 모델링 수식에서 조이스틱의 입력값을 기반으로 항공기의 상태 정보를 연산하여 생성된 정보를 영상 발생 장치로 전송한다. 호스트로 부터 전송된 항공기 상태 정보를 바탕으로 HAT와 HOT를 검증한다.

그림 4은 예천 공항 이륙 전 활주소에 정렬 상



그림 4. 예천 공항 이륙 전

Fig. 4 Before take off from Yecheon airport

HAT/HOT			
CG_HAT	0.0100021362214675	CG_HOT	108.94920349122
FG_HAT	1.96209113535727	FG_HOT	108.959203492084
L.MG_HAT	1.91928013622147	L.MG_HOT	108.94920349122
R.MG_HAT	1.90928013628709	R.MG_HOT	108.959203491154
Aircraft State			
LAT	36.6318473815918	PITCH	0
LONG	128.355545043945	ROLL	0
ALT	108.959205627441	HEAD	447.0102

그림 5. HAT와 HOT 상태 정보

Fig. 5 HAT/HOT state information

태이다. 시뮬레이션에서 HAT 연산 시 앞서 언급한 항공기의 현재 고도와 지형데이터와의 연산을 통해 구한다. 따라서 실세계의 활주로 고도 정보를 이용하여 시뮬레이션 상에서 활주로의 정렬할 경우 HAT가 0이 표시된다. 그림 5에서와 같이 무게 중심점에서의 HAT 정보가 0.01 미터로 활주로의 착륙 장치는 밀착되어 있는 상태로 간주한다. 연산된 HAT와 HOT 정보 확인은 그림 5의 GUI에서 확인할 수 있다. 이러한 GUI에서 윗부분의 HAT/HOT는 네 지점의 HAT와 HOT 정보를 의미하며 아랫부분의 Aircraft State는 호스트 컴퓨터로부터 생성된 현재 항공기의 상태 정보를 의미하며 위도, 경도, 고도와 3축 움직임에 대한 정보를 의미한다.

또한 항공기의 무게 중심점보다 아래쪽에 위치하므로 CG 점의 고도 정보보다 큰 값으로 표시된다. 이밖에 항공기는 지형 아래 부분으로 들어가지 않도록 구현한다. 이를 위해 항공기 무게 중심점 HAT를 바탕으로 HAT 값이 0과 1사이의 값이 되면 고도가 더 이상 내려가지 않도록 구현하고 그림 5에서 현재 고도값은 현재 항공기의 고도 값을 나타내며 CG_HOT값과 오차가 0.01미터를 나타내고 있다. 또한 HAT 정보가 오차 범위 이내 상황에서 검증에 의한 지형과의 충돌 시 그림 6과 같이 지면과 충돌 경고 메시지를 표시한다.

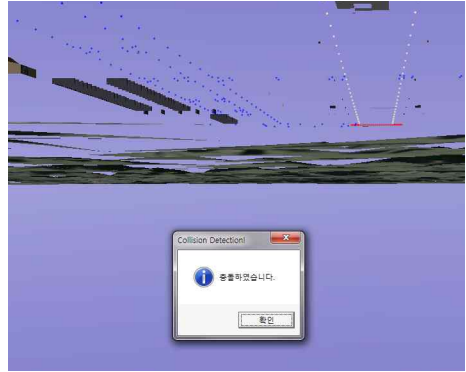


그림 6. 오차범위 이내 HAT에서의 이벤트 처리

Fig. 6. Collision detection event handling

V. 결 론

본 논문에서는 항공기 시뮬레이터 개발 기술 중 탑승자의 지형 충돌 경고와 이착륙 훈련 시 정밀도 높은 고도 감지를 위한 HAT와 HOT를 구현하는 방법을 제안하였다. 구현과 검증을 위해 호스트 컴퓨터에서 연산되어 전송된 고도 정보를 기반으로 OSG를 이용하여 영상 발생 장치 소프트웨어를 개발하였다. 구현된 HAT와 HOT 정보는 예천 공항 활주로 정렬 상태에서 HAT 정보가 0이 되는지를 확인하였고 호스트로부터 전송된 고도 정보와 비교하여 구현된 HAT와 HOT 정보의 정확성을 확인하였다.

향후 HAT 구현 경험을 바탕으로 LOS(Line Of Sight)을 연구하여 통신 가시선 분석과 기타 충돌 경고 항로선 단면도 분석, 그리고 항공기 시뮬레이션 상황에 필요한 기능에 관한 연구를 추후 진행할 예정이다. 추가적으로 현재 개발되고 있는 영상 발생 장치 소프트웨어의 기능 확장을 통하여 실제 상용화할 계획이다.

References

- [1] S.J Yoon, Simulation and Simulator, Sunhak Publishing Corporations, 2003 (in Korean).
- [2] G.C. Miliareisis, C.V.E. Paraschou, "Vertical accuracy of the SRTM DTED level 1 of Crete," International Journal of Applied Earth Observation and Geoinformation, Vol. 7, No. 1, pp.49-59, 2005.
- [3] S. Eidenbenz, C. Stamm, P. Widmayer,

“Positioning Guards at Fixed Height above a Terrain - an Optimum Inapproximability Result,” Lecture Notes in Computer Science, pp.187-198, 1998.

- [4] Flight Simulator X SDK Terrain and Scenery - Elevations ,<http://msdn.microsoft.com/en-us/library/cc707102.aspx#Elevations>
- [5] Flight Gear Developer Guide, <http://wiki.flightgear.org/Portal:Developer>
- [6] X-Plane SDK Object PluginsAndObjects, <http://www.xsquawkbox.net/xpsdk/mediawiki/PluginsAndObjects>
- [7] J.H. Kim, J.S. Seo, C.H. Park, J. Hwang, H.D. Ko, “Development of Scene Graph Library for Mobile Platfoms,” Journal of Korea Multimedia Society, Vol. 13, No. 5, pp.792-801, 2010 (in Korean).
- [8] R. Wang, X. Qian, OpenSceneGraph 3 Cookbook, Packt Publishing Ltd., 2010.
- [9] R. Wang, X. Qian, OpenSceneGraph 3 Beginner’s Guide, Packt Publishing Ltd., 2012.

저 자 소 개

이충재



2009년 한서대학교 항공 전자 시뮬레이션학과 학사.

2012년 경상대학교 항공 우주특성화대학과 석사.

현재, 경상대학교 정보과학과 박사과정.

관심분야: 항공 시뮬레이션 소프트웨어.

Email: cjlee@gnu.ac.kr

김종범



2012년 경상대학교 컴퓨터 교육학과 학사.

현재, 경상대학교 정보과학과 석사과정.

관심분야: 항공 시뮬레이션 소프트웨어.

Email: karma15513@nate.com

김기일



2002년 충남대학교 컴퓨터과학과 석사.

2005년 충남대학교 컴퓨터과학과 박사.

현재, 경상대학교 정보과학과 부교수.

관심분야: 네트워크&항공SW, 프로토콜 설계.

Email: kikim@gnu.ac.kr